

AI-Driven Custom Home Design Assistant

AI-Driven Custom Home Design Assistant

The Custom Home Design Creator project focuses on developing an AI-powered tool for generating personalized home design plans. The objective is to build a generative model that creates custom home layouts based on user inputs such as preferences, spatial requirements, and architectural styles. By analyzing these inputs, the model produces detailed and aesthetically pleasing home designs that align with the user's vision and functional needs. This tool aims to simplify the home design process, allowing users to explore and visualize their ideal living spaces efficiently, while providing a creative and personalized approach to home planning and design.

Scenario 1: Real Estate Development

In a real estate development firm, the goal is to offer potential buyers customized home designs based on their preferences. Clients input their desired features, such as the number of bedrooms, architectural style, and special amenities. The Custom Home Design Creator generates personalized home layouts that match these specifications. This approach helps clients visualize their ideal homes and enhances their purchasing experience, similar to how tailored property showcases can drive interest and sales.

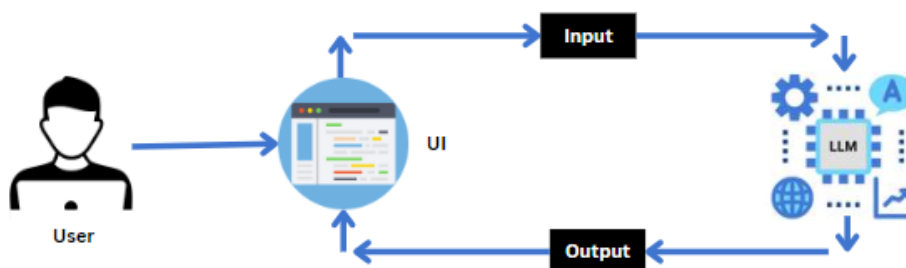
Scenario 2: Home Renovation Services

For a home renovation company, the tool is used to create design proposals for clients looking to remodel their existing homes. Users provide details about their current space, renovation goals, and style preferences. The Custom Home Design Creator then generates updated design plans that reflect these inputs, helping clients visualize potential changes. This facilitates the decision-making process and allows for more informed planning, akin to how virtual staging is used in real estate.

Scenario 3: Architectural Firm

In an architectural firm, the tool is utilized to quickly generate preliminary design concepts for clients based on their input. Architects and clients collaborate by specifying requirements such as room layout, design aesthetics, and functional needs. The Custom Home Design Creator produces detailed design drafts that can be refined further. This accelerates the design process and provides a clear starting point for discussions, similar to how initial sketches are used in architectural planning.

Architecture:



AI-Driven Custom Home Design Assistant

Project Flow

- 1. User Input via Streamlit UI:
 - Users input a prompt (e.g., topic, keywords) and specify parameters such as the desired length, tone, or style through the Streamlit interface.
 3. Backend Processing with Generative AI Model:
 - The input data is sent to the backend, where it interfaces with the selected Generative AI model (e.g., GPT-4, Gemini, etc.).
 - The model processes the input, generating text based on the specified parameters and user input.
 6. Content Generation:
 - The AI model autonomously creates content tailored to the user's specifications. This could be a blog post, poem, article, or any other form of text.
 8. Return and Display Generated Content:
 - The generated content is sent back to the frontend for display on the Streamlit app.
 - The app presents the content to the user in an easily readable format.
 11. Customization and Finalization:
 - Users can further customize the generated content through the Streamlit UI if desired. This might include editing text, adjusting length, or altering tone.
 13. Export and Usage:
 - Once satisfied, users can export or copy the content for their use, such as saving it to a file or directly sharing it.

Requirements Specification

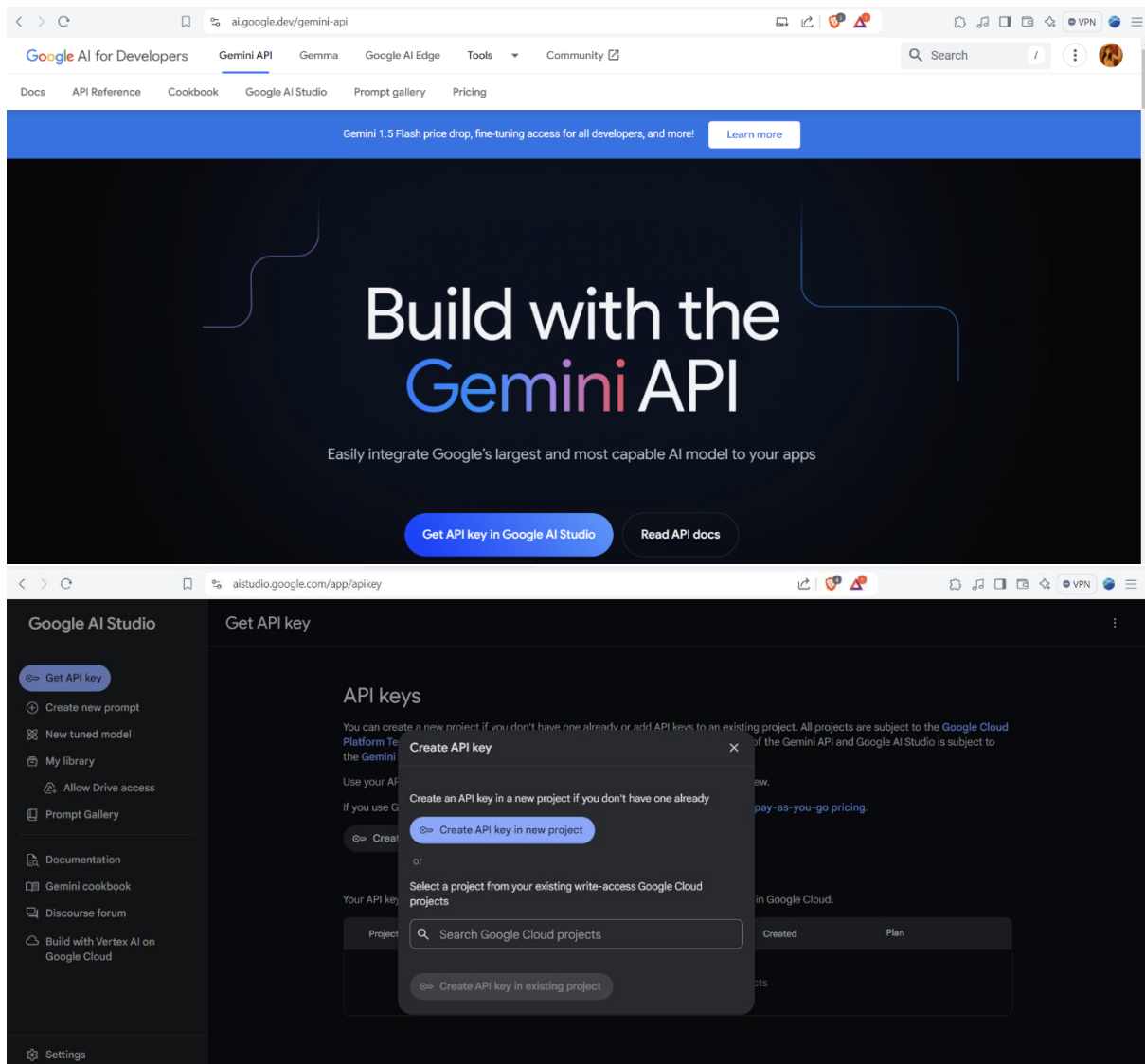
Install the libraries

- `pip install streamlit`
- `pip install google.generativeai`

Initializing the Models

Generating API

AI-Driven Custom Home Design Assistant



Link: <https://ai.google.dev/gemini-api>

Enable the Gemini API

- Once your project is created, navigate to the API & Services Dashboard.
- Click on Enable APIs and Services at the top.
- Search for "Gemini API" in the API library.
- Select the Gemini API and click Enable.

Initialize the pre-trained model

-
-
- Streamlit, a popular Python library, is imported as `st`, enabling the creation of user interfaces directly within the Python script.

AI-Driven Custom Home Design Assistant

- Google Generative AI (genai): Imported to interact with the Gemini Pro model.

Activity 2.1: Importing Libraries

```
import streamlit as st
```

```
import google.generativeai as genai
```

1. Streamlit (streamlit as st): This is a popular open-source framework used for creating web apps with Python. The st alias allows you to easily access Streamlit's functions for building user interfaces, like adding buttons, displaying data, and more.

2. Google's Generative AI (google.generativeai as genai): This module likely provides access to Google's generative AI models. The genai alias allows you to interact with these models to generate text, images, or other outputs, depending on what Google's generative AI API offers.

Activity 2.2: Configuring of the Gemini Pro API

```
# Configure Generative AI
```

```
genai.configure(api_key=API_KEY)
```

This code sets an API key for accessing Google's Generative AI services. The api_key variable stores the key, and genai.configure(api_key=api_key) configures the genai module to use this key for authenticating requests to Google's Generative AI API. This allows secure interaction with the AI models.

```
import google.generativeai as genai
```

```
import os
```

```
from dotenv import load_dotenv
```

```
# Load API Key from .env
```

```
load_dotenv()
```

```
API_KEY = os.getenv("AlzaSyBV_VcW7qu8YVew0BIVRNLB3p96E-4h2L8")
```

```
# Configure Generative AI
```

```
genai.configure(api_key=API_KEY)
```

```
# Function to generate home design ideas
```

AI-Driven Custom Home Design Assistant

```
def generate_design_idea(style, size, rooms):
```

```
    return f"The design for a {style} home of {size} with {rooms} rooms includes modern layouts and materia
```

Interfacing with Pre-trained Model

Creating Function & Defining the model

Function to generate home design ideas

```
def generate_design_idea(style, size, rooms):
```

```
    return f"The design for a {style} home of {size} with {rooms} rooms includes modern layouts and materials."
```

This function, `generate_design_idea`, is designed to create a custom home design plan based on user inputs. It takes three parameters:

- `style`: The style of the home design (e.g., Modern, Rustic).
- `size`: The size of the home in square feet or another unit.
- `rooms`: The number of rooms in the home.

`model_name`: Specifies which AI model to use (in this case, "gemini-1.5-pro").

`generation_config`: A dictionary of settings that guide how the model generates the text (e.g., temperature, top_p, etc.).

import requests

```
def fetch_image_from_lexica(style):
```

```
    lexica_url = f"https://lexica.art/api/search?q={style}"
```

```
    response = requests.get(lexica_url)
```

```
    if response.status_code == 200:
```

```
        data = response.json()
```

```
        if "images" in data and len(data["images"]) > 0:
```

```
            return data["images"][0]["src"]
```

```
    return None # Return None if no images are found
```

Context Definition:

AI-Driven Custom Home Design Assistant

- A context string is defined to specify the requirements for generating a custom home design plan. It includes details such as style, size, and number of rooms. It also outlines additional aspects to include, like layout suggestions, color schemes, and furniture recommendations, and requests the response in Markdown format.

Chat Session Initialization:

- A chat session is started with a model (e.g., an AI model) using the `start_chat` method.
- The `history` parameter is used to provide initial input to the model. In this case, it consists of a single message from the user that includes the context for the conversation.
- This setup provides the model with the necessary information to generate a response based on the specified requirements.

```
• return response.text if response else "No response generated."
```

Send Message:

- A message containing the context is sent to the chat session, which requests the model to generate a response based on the provided information.

Process Response:

- The code checks the format of the response content. If it's a straightforward string, it assigns it directly to a variable.
- If the response content is more complex (e.g., a structured object), it extracts the relevant text from the first part of the content.

Return Text:

- The extracted or processed text is returned, which represents the model's generated response.

Activity 2: Creating Function for image

Function to generate home design ideas

```
def generate_design_idea(style, size, rooms):
```

```
    model = genai.GenerativeModel("gemini-1.5-pro")
```

```
    prompt = f"Create a custom home design plan for a {style} home, {size} sq ft, with {rooms} rooms."
```

```
    response = model.generate_content(prompt)
```

```
    return response.text if response else "No response generated."
```

Function Definition:

AI-Driven Custom Home Design Assistant

- The function `fetch_image_from_lexica` is designed to retrieve an image from Lexica.art based on a specified design style.

Construct URL:

- `lexica_url` is created by embedding the style parameter into the URL. This URL is used to query the Lexica.art API for images that match the given design style.

Send Request:

- A GET request is sent to the Lexica.art API using the constructed URL. This request fetches data from the API.

Parse Response:

- The response from the API is converted from JSON format to a Python dictionary.

Check and Return Image URL:

- The function checks if there are any images in the response. If images are present, it returns the URL of the first image.
- If no images are found, it returns `None`.

• Model Deployment

- In this milestone, we deploy the created model using Streamlit. Streamlit allows us to create a user-friendly web interface, enabling users to interact with the model through their web browser

Starting Streamlit

```
# Streamlit UI
```

```
st.title("Custom Home Design Assistant")
```

```
style = st.text_input("Enter Home Style (e.g., Modern, Rustic)")
```

```
size = st.text_input("Enter Home Size (sq ft)")
```

```
rooms = st.text_input("Enter Number of Rooms")
```

```
if st.button("Generate Design"):
```

```
    if style and size and rooms:
```

```
        design_idea = generate_design_idea(style, size, rooms)
```

```
        st.markdown(design_idea)
```

```
    else:
```

AI-Driven Custom Home Design Assistant

```
st.warning("Please fill in all fields.")
```

Title:

- Displays the title of the Streamlit app, "Custom Home Design Assistant."

Text Input for Style:

- Provides a text input field for users to enter the design style of the home.

Text Input for Size:

- Provides a text input field for users to enter the size of the home.

Text Input for Number of Rooms:

- Provides a text input field for users to enter the number of rooms in the home.

Activity 2: Displaying for user

Submit Button: `if st.button("Generate Design"):`

`if style and size and rooms:`

```
    design_idea = generate_design_idea(style, size, rooms)
```

```
    st.markdown(design_idea)
```

`else:`

```
    st.warning("Please fill in all fields.")
```

- A button labeled "Generate Design" is displayed to the user.

Button Action:

- When the button is clicked, the code checks if all input fields (style, size, and rooms) are filled.

Generate Design Idea:

- If all fields are filled, it calls a function to generate a design idea based on the provided inputs.

Fetch Image:

AI-Driven Custom Home Design Assistant

- It then fetches an image related to the design style from Lexica.art.

Display Design Idea:

- The generated design idea is displayed in the app using Markdown.

Display Image:

- If an image URL is returned, the image is displayed with a caption. If no image is found, a warning message is shown instead.

Field Validation: # Streamlit UI

```
st.title("Custom Home Design Assistant")
```

```
style = st.text_input("Enter Home Style (e.g., Modern, Rustic)")
```

```
size = st.text_input("Enter Home Size (sq ft)")
```

```
rooms = st.text_input("Enter Number of Rooms")
```

```
if st.button("Generate Design"):
```

```
    if style and size and rooms:
```

```
        design_idea = generate_design_idea(style, size, rooms)
```

```
        st.markdown(design_idea)
```

```
    else:
```

```
        st.warning("Please fill in all fields.")
```

- If any of the input fields are empty, a warning message prompts the user to fill in all fields.
- **Displaying for user**
-
- Submit Button:
- A button labeled "Generate Design" is displayed to the user.
- Button Action:
- When the button is clicked, the code checks if all input fields (style, size, and rooms) are filled.
- Generate Design Idea:
- If all fields are filled, it calls a function to generate a design idea based on the provided inputs.

AI-Driven Custom Home Design Assistant

- Fetch Image:
- It then fetches an image related to the design style from Lexica.art.
- Display Design Idea:
- The generated design idea is displayed in the app using Markdown.
- Display Image:
- If an image URL is returned, the image is displayed with a caption. If no image is found, a warning message is shown instead.
- Field Validation:
- If any of the input fields are empty, a warning message prompts the user to fill in all fields.

• Running the web application

-
- PS C:\Users\HOME DECOR AI> streamlit run app.py
-
- You can now view your Streamlit app in your browser.
-
- Local URL: <http://localhost:8507>
- Network URL: <http://192.168.40.180:8507>
-
- The application is now running and can be accessed locally through the provided URL. It is also available on the network via the network URL, allowing access from other devices on the same network.

• Output

- **Custom Home Design Assistant**
- Enter Home Style (e.g., Modern, Rustic)
- Enter Home Size (sq ft)
- Enter Number of Rooms
- Generate Design
- **Rustic Home Design Plan (3000 sqft, 3 Bedrooms)**
- This plan focuses on creating a spacious, open-concept rustic home with three large bedrooms, emphasizing natural materials and a warm, inviting atmosphere.
- **Layout Suggestions:**
- **Open-Concept Living:** Combine the kitchen, dining, and living areas into one large space. This maximizes the feeling of spaciousness and encourages interaction. A large stone fireplace could act as a focal point in the living area.

AI-Driven Custom Home Design Assistant

- **Master Suite Wing:** Designate one wing of the house for the master suite, offering privacy and incorporating a large bedroom, walk-in closet, and ensuite bathroom with a soaking tub and separate shower.
- **Guest Bedrooms:** Place the other two bedrooms on the opposite side of the house, potentially sharing a Jack-and-Jill bathroom or having separate access to a hallway bathroom. Consider making one bedroom slightly larger and outfitting it as a potential office/guest room combination.
- **Outdoor Living:** Dedicate a substantial portion of the square footage to a covered porch or patio, blurring the lines between indoor and outdoor living. This could include an outdoor kitchen and dining area for entertaining.
- **Mudroom/Entryway:** Include a dedicated mudroom or entryway area to manage outdoor gear and prevent clutter from entering the main living space.
- **High Ceilings:** Utilize high, vaulted ceilings, particularly in the living area, to enhance the sense of spaciousness and showcase exposed wooden beams.
- **Color Schemes:**
 - **Walls:** Warm, earthy tones like beige, cream, sage green, or light gray. Consider using textured plaster or shiplap for added rustic charm.
 - **Trim:** Darker wood tones like walnut or mahogany to contrast with the lighter walls.
 - **Accents:** Pops of color can be introduced through textiles, artwork, and accessories. Consider using deep reds, blues, greens, or oranges inspired by nature.
- **Flooring:** Wide-plank hardwood flooring in a natural finish or stained a warm brown. Stone or slate could be used in the entryway and bathrooms.
- **Furniture Recommendations:**
 - **Living Area:** Oversized, comfortable sofas and armchairs upholstered in leather or linen. A large wooden coffee table, rustic side tables, and a statement area rug.
 - **Dining Area:** A large farmhouse-style dining table with matching chairs or benches. Consider a sideboard or buffet for storage and display.
 - **Kitchen:** Custom cabinetry with a distressed finish. Open shelving to display pottery and cookware. A large kitchen island with seating. Consider incorporating stainless steel appliances for a modern rustic touch.
 - **Bedrooms:** Wooden bed frames with upholstered headboards. Nightstands with drawers for storage. Dressers and chests of drawers in a matching style. Consider incorporating a seating area in the master bedroom.
 - **Bathrooms:** Vanities with stone countertops. Freestanding soaking tubs. Walk-in showers with rainfall showerheads. Wrought iron light fixtures.
- **Material Palette:**
 - **Wood:** Exposed beams, wide-plank flooring, reclaimed wood accents.

AI-Driven Custom Home Design Assistant

- **Stone:** Fireplace surround, countertops, accent walls.
- **Metal:** Wrought iron light fixtures, hardware, and accents.
- **Textiles:** Linen, wool, cotton, and leather in natural textures and colors.
- **Example Room Breakdown (Approximate):**
 - Open-concept living/dining/kitchen: 1500 sqft
 - Master Suite (bedroom, bathroom, closet): 700 sqft
 - Bedroom 2: 300 sqft
 - Bedroom 3/Office: 300 sqft
 - Mudroom/Entryway/Hallways/Bathroom: 200 sqft
- This plan provides a starting point. Consult with an architect and interior designer to tailor the design to your specific needs and preferences and create a truly unique and personalized rustic home.

• Conclusion

- The Travel Itinerary Planner project showcases the transformative impact of AI on travel planning. By harnessing Google Generative AI and integrating it with Streamlit, the tool offers a sophisticated and user-friendly solution for crafting personalized travel itineraries. The ability to input destination, trip duration, and activity preferences results in tailored travel plans that include detailed daily schedules, sightseeing opportunities, and dining options. This project not only streamlines the travel planning process but also enhances the overall travel experience by delivering well-organized and customized recommendations. The successful implementation of AI in this context underscores its potential to revolutionize personal planning across various domains.